BANK CUSTOMER CHURN

Group 5

PRESENTED BY:

Anya Rajan
Caomengyu Xue
Chuyue Cui
Priyanshi Gajjar
Ria Kalluri
Soo Young Lee

March 10th, 2025

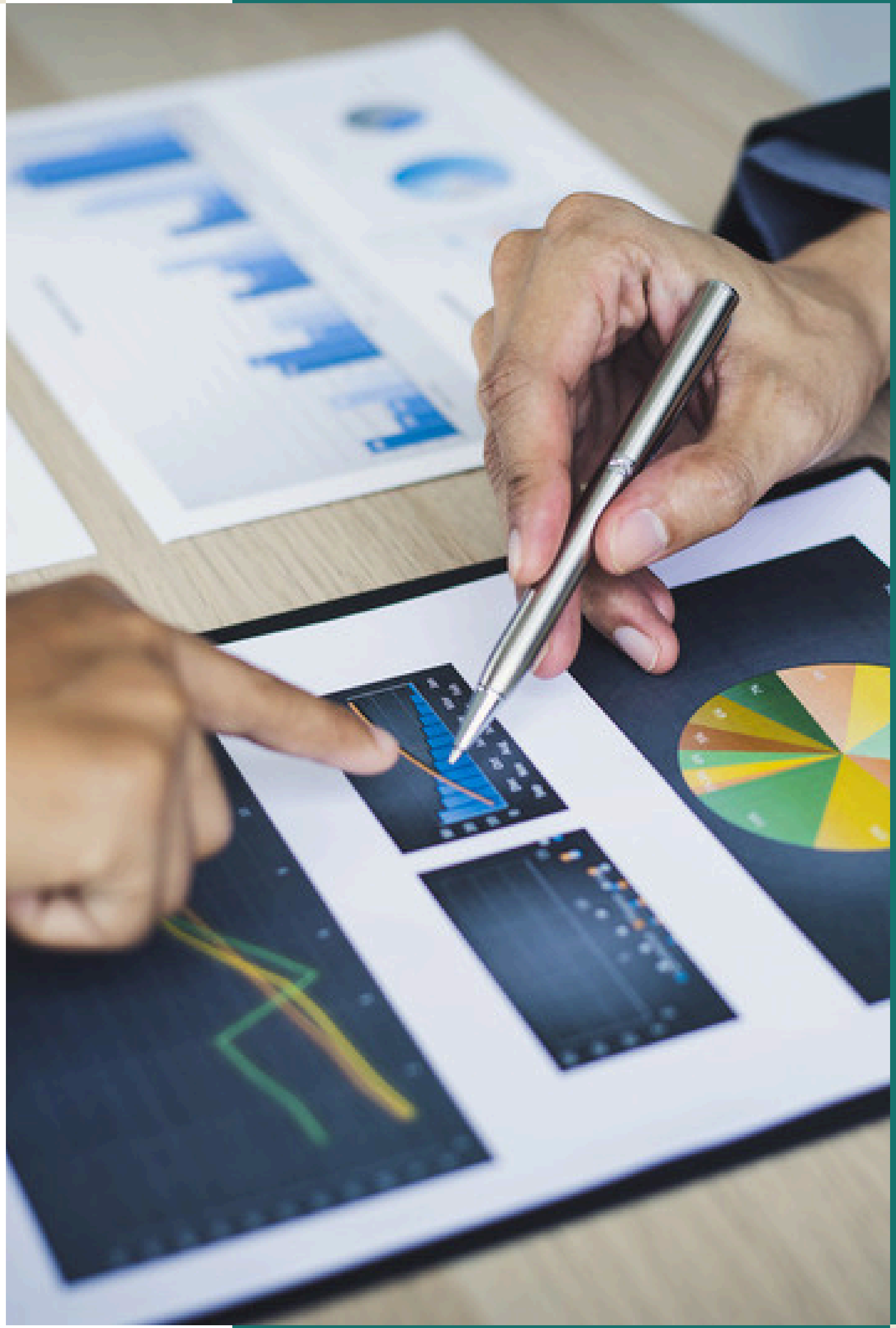
TABLE OF CONTENTS

BUSINESS PROBLEM

WHY DOES CUSTOMER CHURN PREDICATION MATTER?

- Business Impacts of Customer Churn:
  - Revenue Loss
  - High Cost of Customer Acquisition
  - Brand Reputation & Market Position
  - Customer Lifetime Value
  - Investor & Business Confidence

- Why Customer Churn Analysis is Important:
  - Helps banks identify which customers at risk of leaving
  - Allows for optimized retention strategies
  - Reduced unnecessary retention costs by focusing on high-risk customers
  - Improves customer satisfaction & long-term loyalty



DATA DESCRIPTION

- Source : Kaggle - Bank Customer Churn Dataset
- Size : 10,000 customer records
- Features : Numerical & Categorical
- Target Variable : ‘ Exited ‘ (1 = Churned, 0 = Retained)

```
Data columns (total 18 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   RowNumber           10000 non-null  int64
 1   CustomerId          10000 non-null  int64
 2   Surname             10000 non-null  object
 3   CreditScore         10000 non-null  int64
 4   Geography           10000 non-null  object
 5   Gender              10000 non-null  object
 6   Age                 10000 non-null  int64
 7   Tenure              10000 non-null  int64
 8   Balance             10000 non-null  float64
 9   NumOfProducts       10000 non-null  int64
 10  HasCrCard           10000 non-null  int64
 11  IsActiveMember      10000 non-null  int64
 12  EstimatedSalary     10000 non-null  float64
 13  Exited              10000 non-null  int64
 14  Complain            10000 non-null  int64
 15  Satisfaction Score  10000 non-null  int64
 16  Card Type           10000 non-null  object
 17  Point Earned        10000 non-null  int64
dtypes: float64(2), int64(12), object(4)
memory usage: 1.4+ MB
```

EXPLORATORY DATA ANALYSIS

Missing values

```
RowNumber            0
CustomerId           0
Surname              0
CreditScore          0
Geography            0
Gender               0
Age                  0
Tenure               0
Balance              0
NumOfProducts        0
HasCrCard            0
IsActiveMember       0
EstimatedSalary      0
Exited               0
Complain             0
Satisfaction Score   0
Card Type            0
Point Earned         0
dtype: int64
```

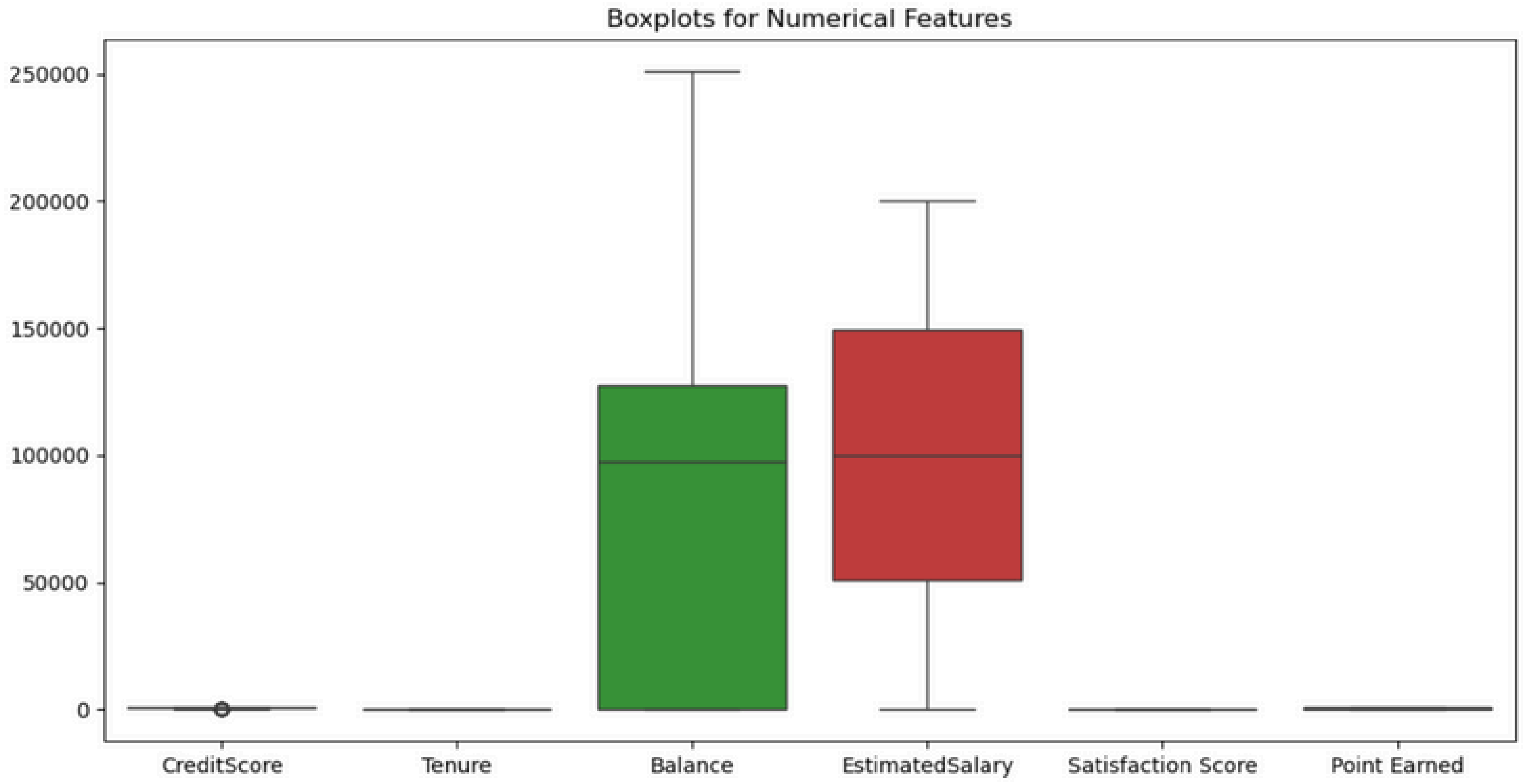
Outliers

```
CreditScore          0
Tenure               0
Balance              0
EstimatedSalary      0
Satisfaction Score   0
Point Earned         0
dtype: int64
```

Duplicates

```
Number of duplicate rows: 0
```

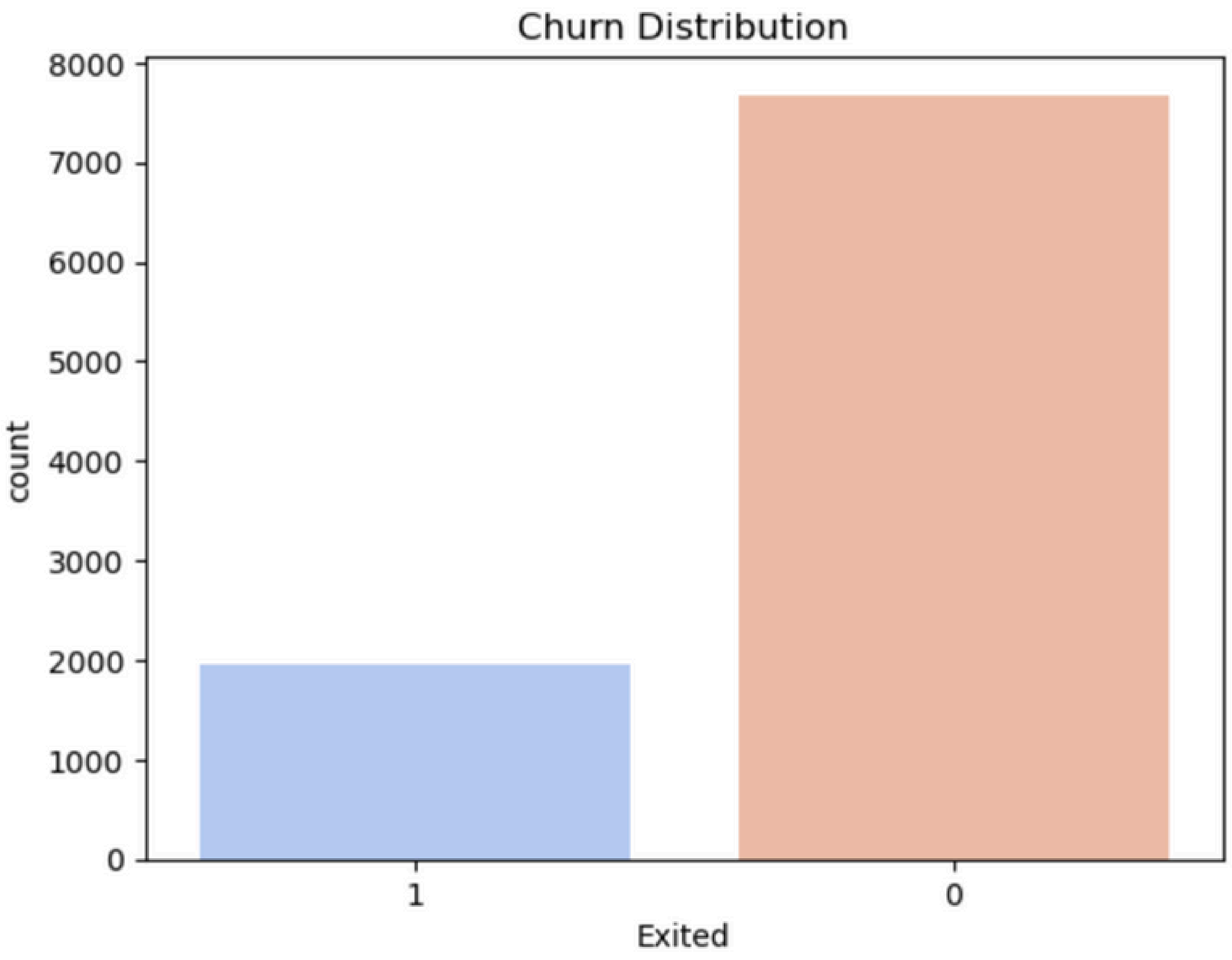
Boxplots for Outlier Detection



- **Balance** has a large range but no significant outliers, indicating a reasonable distribution.
- **EstimatedSalary** is widely distributed with no outliers, aligning with expectations.

VISUALIZATION

TARGET VARIABLE (CHURN) DISTRIBUTION

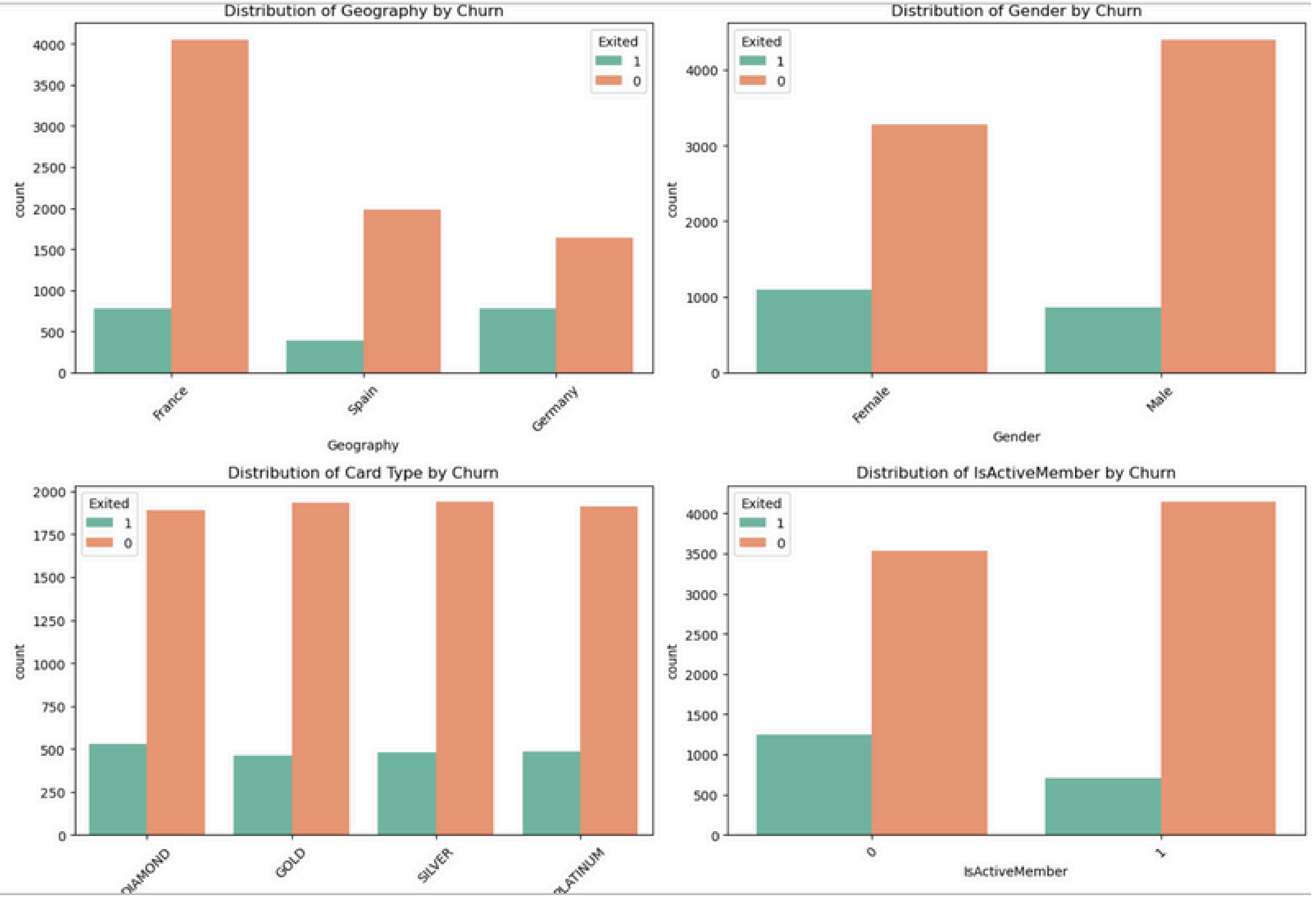


Churn Rate: 20.26%

- Approximately **20%** of customers lost (Exited = 1) and **80%** retained (Exited = 0)
- The number of customers that do not lose is far more than those that lose, and there is a category imbalance in data distribution.

VISUALIZATION

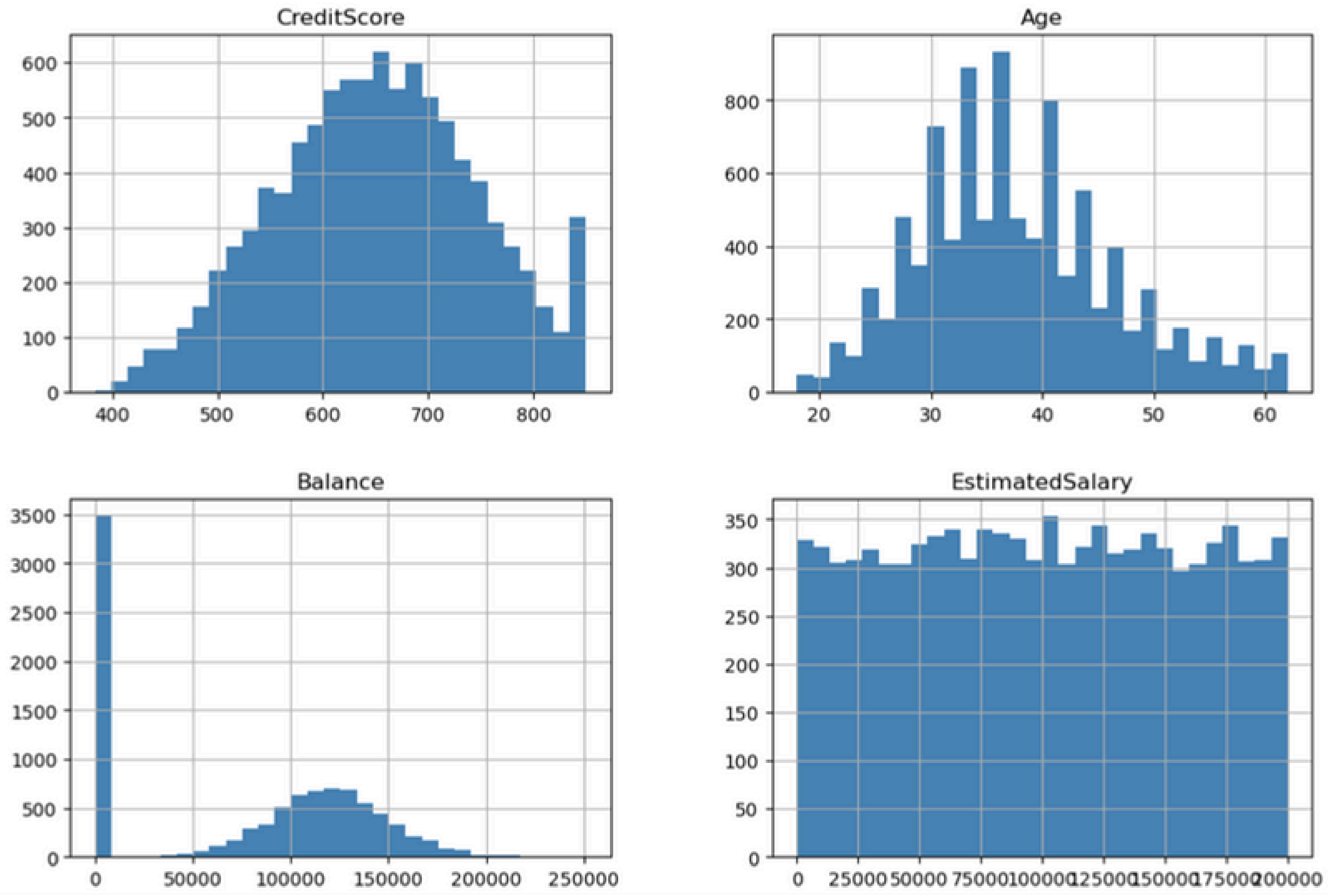
CATEGORICAL FEATURE DISTRIBUTION



- **Geography:** Higher churn in Germany; France and Spain show better retention.
- **Gender:** Women churn more, possibly due to sensitivity to banking services.
- **Card Type:** Minimal impact on churn. Other factors may matter more.
- **Activity Status:** Active members churn less; inactive ones are more likely to leave.

VISUALIZATION

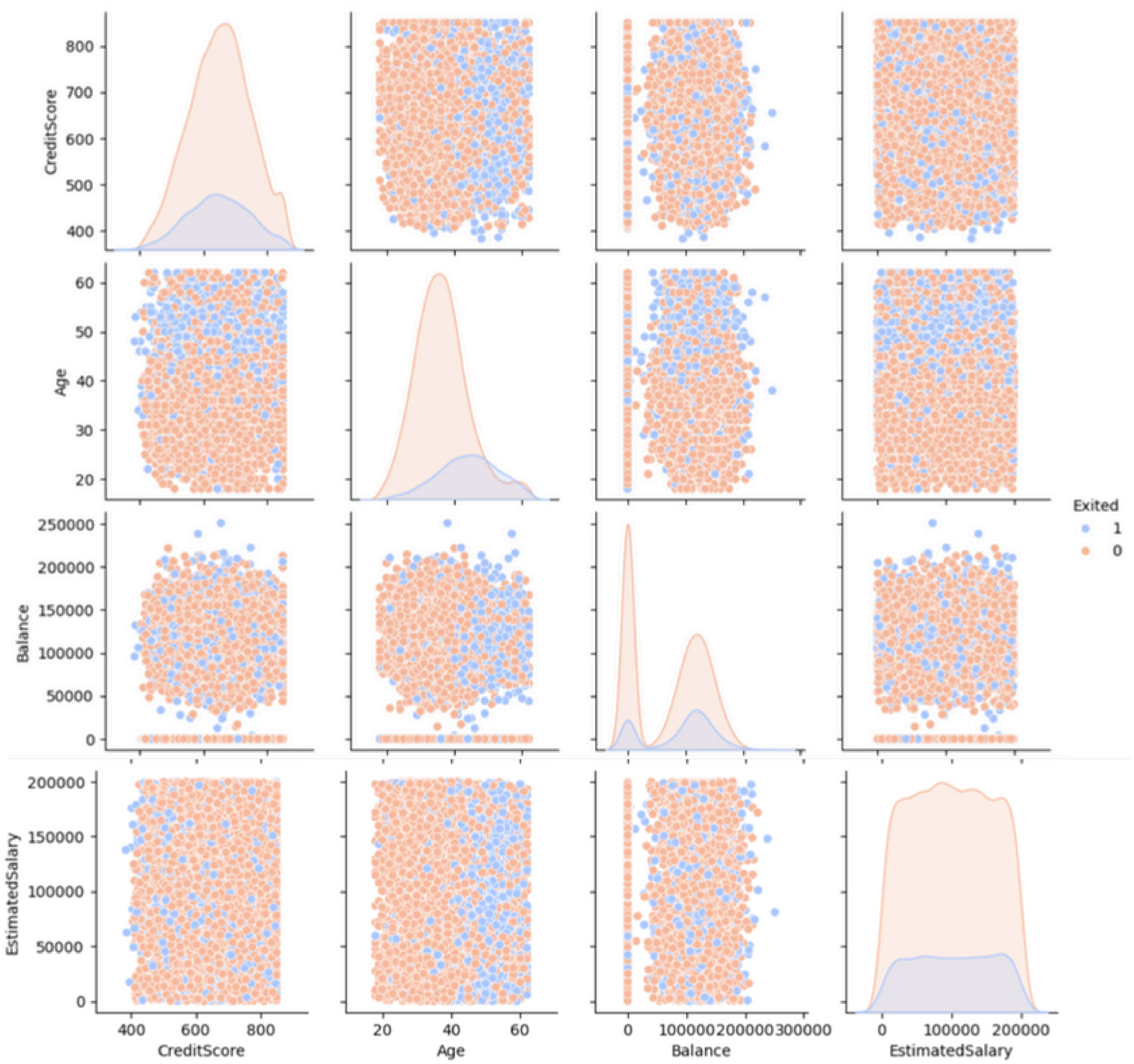
NUMERICAL FEATURE DISTRIBUTION



- **CreditScore:** Normally distributed, mostly between 600-750.
- **Age:** Slight right skew; most are 30-40 years old, with fluctuations in 40-50.
- **Balance:** Many have zero balance; others cluster around 100K-150K.
- **EstimatedSalary:** Evenly distributed, no clear outliers.

VISUALIZATION

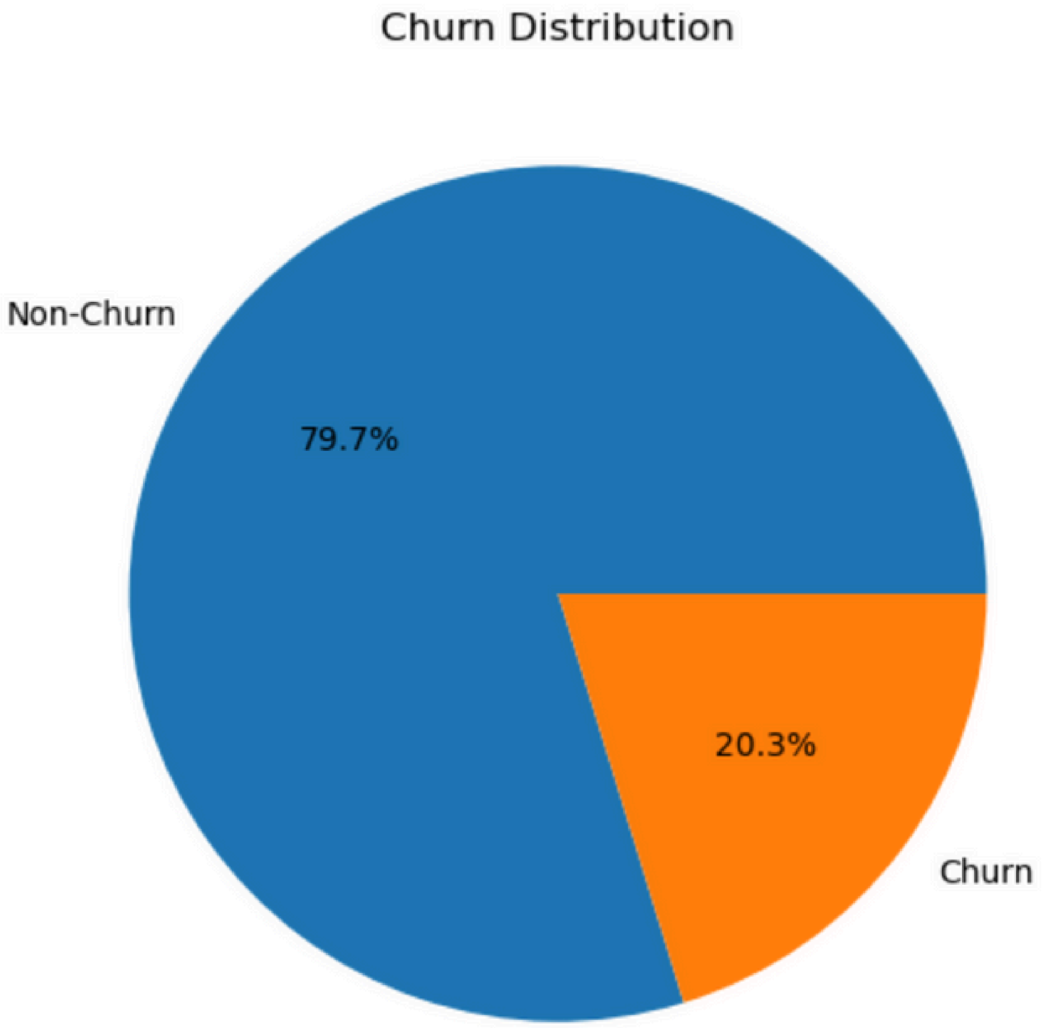
CHURN VS IMPORTANT FEATURES

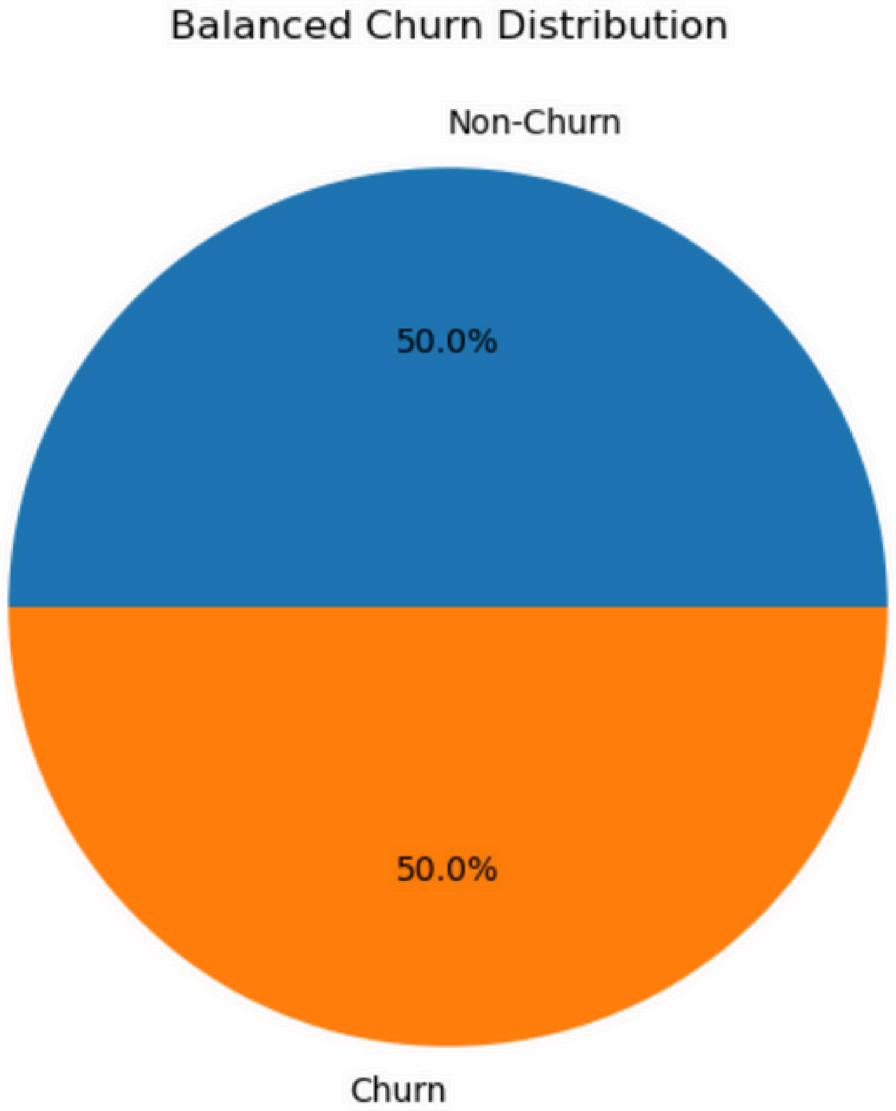


- **Age:** Older customers (especially 50+) have a higher churn rate.
- **Balance:** Customers with zero balance show significant churn, possibly inactive accounts.
- **CreditScore & EstimatedSalary:** No clear pattern with churn, suggesting weak influence.

VISUALIZATION

WHY WE CHOSE BALANCED DATA?






```
Percentage of Non-Churn (0): 79.74%
Percentage of Churn (1): 20.26%
```

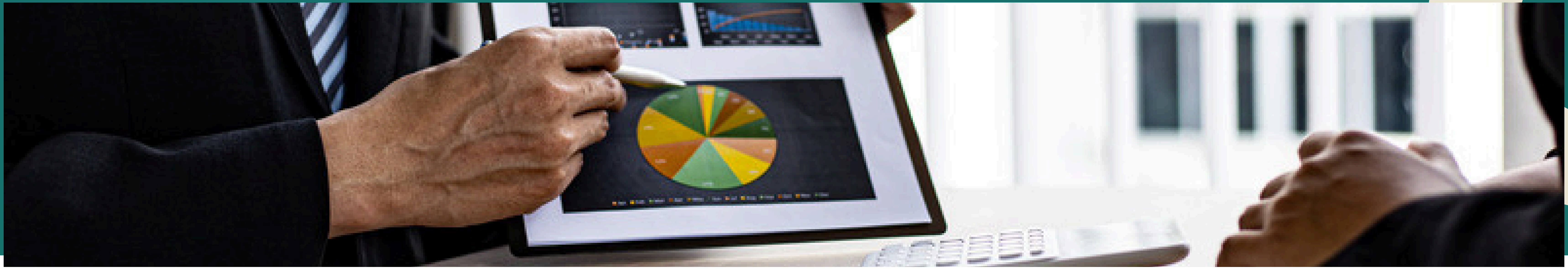
```
Balanced Percentage of Non-Churn (0): 50.00%
Balanced Percentage of Churn (1): 50.00%
```

WHY NOT UNBALANCED DATA?

- Unbalanced data improves accuracy but misses many churned customers.
- Low recall leads to a biased model favoring non-churn customers.
- Customer retention is more affected by missing churned customers than false positives.

Maximize Recall to detect at-risk customers early

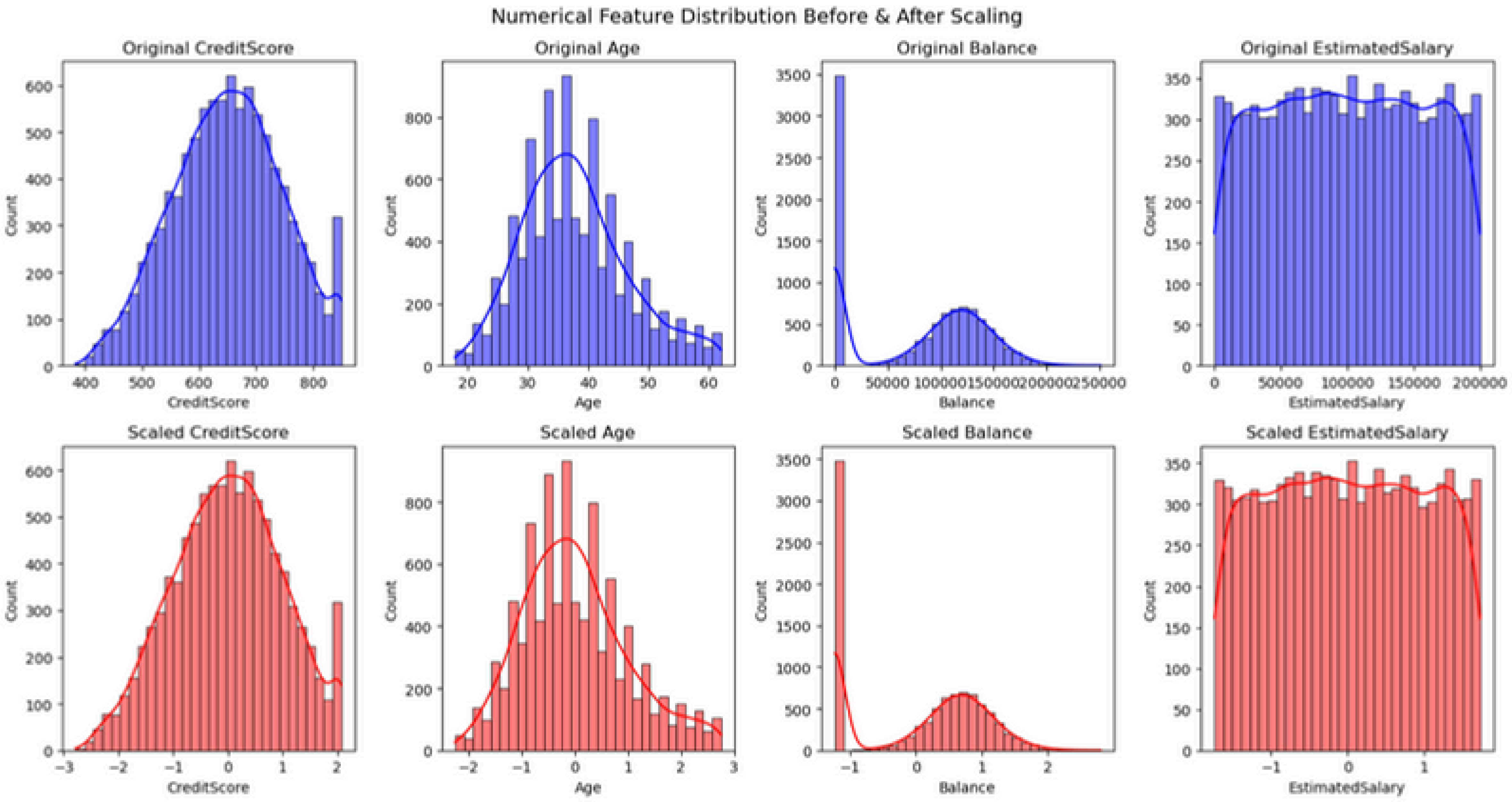
Better identification of churned customers



FEATURE ENGINEERING

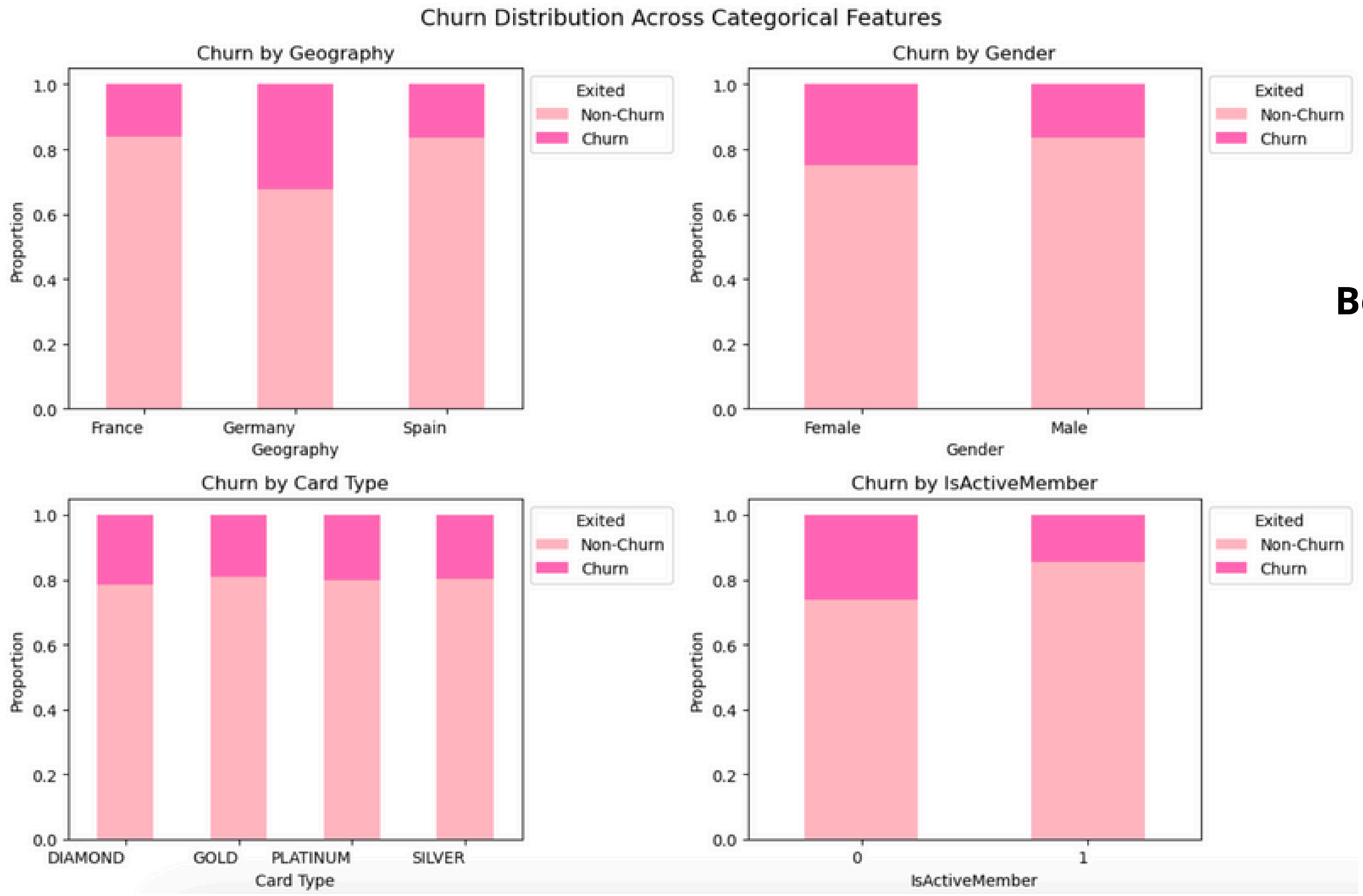
- Numerical features have different value ranges, which can mislead the model.
- Balance is much larger than CreditScore and Age, requiring standardization.
- We applied standard scaling to normalize the data, ensuring fair feature weighting

NUMERICAL FEATURES



FEATURE ENGINEERING

CATEGORICAL FEATURES - BEFORE

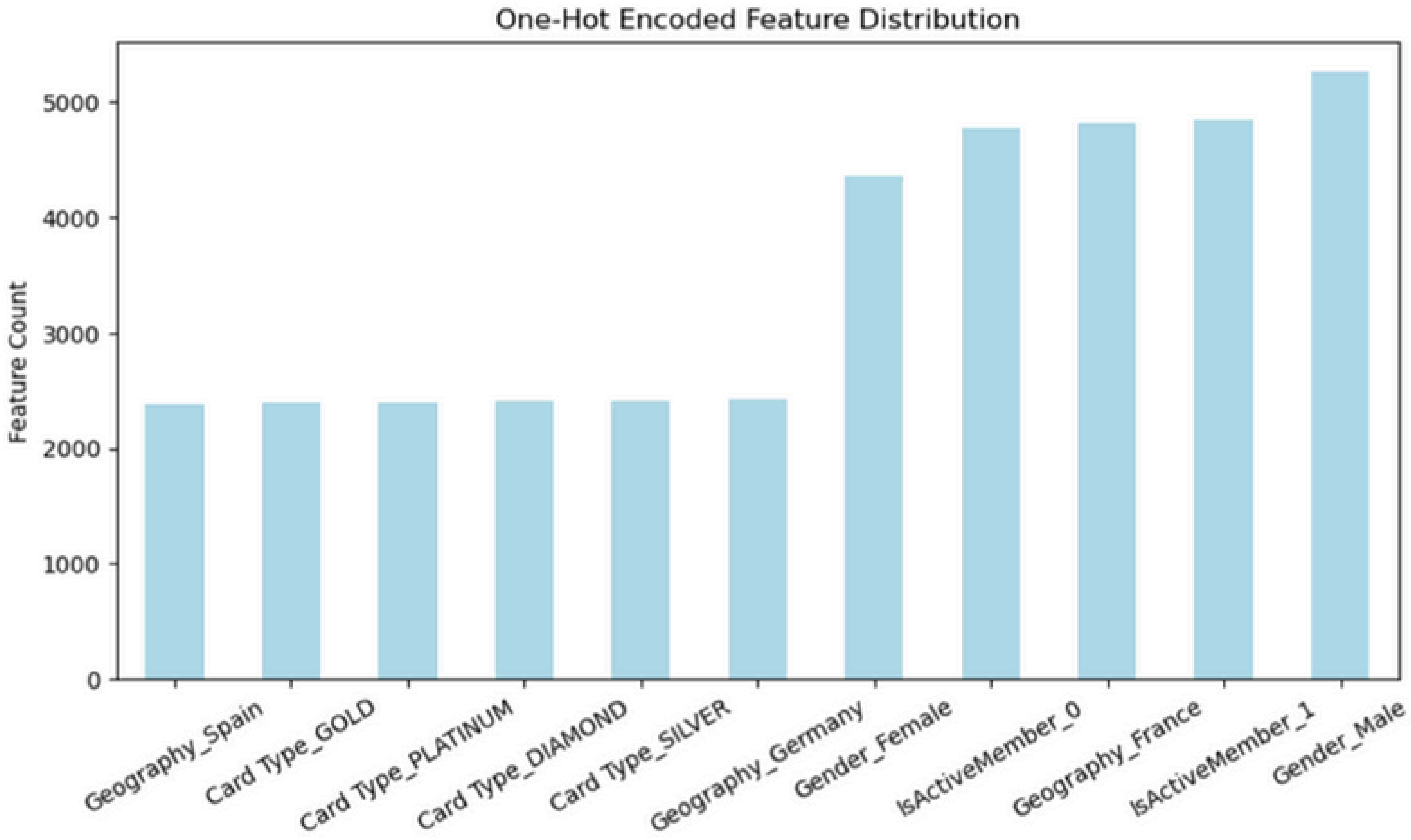


Before Transformation:

- Geography, Gender, and Customer Activity were categorical variables that the model couldn't directly process.
- Certain categories, like inactive members, showed a higher churn rate

FEATURE ENGINEERING

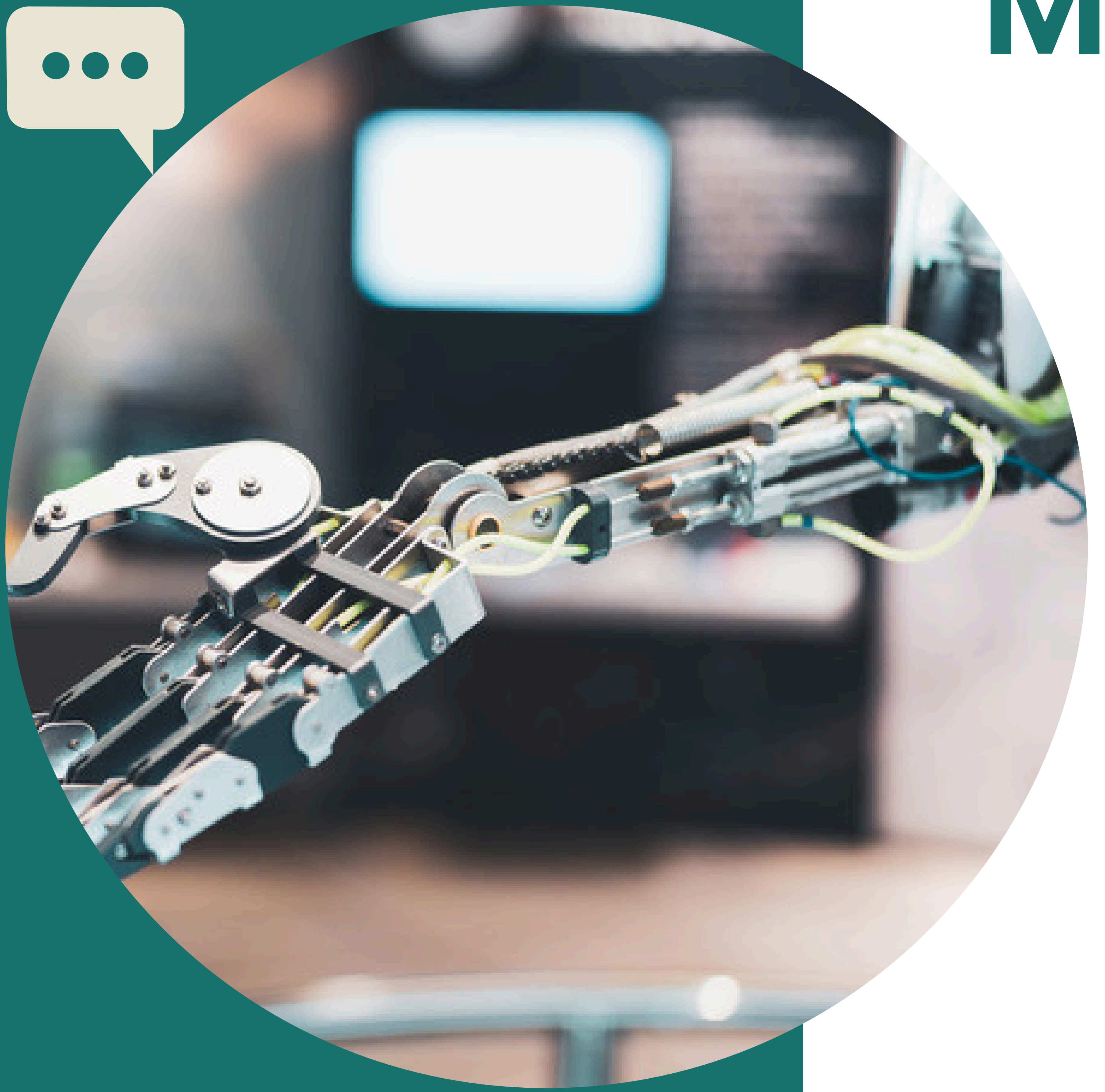
CATEGORICAL FEATURES - AFTER



After Transformation

- Geography --> Geography_Spain、Geography_Germany、Geography_France
- Gender --> Gender_Male and Gender_Female
- Gender_Male、IsActiveMember_1 has more impact

MODEL EVALUATION



1️⃣ Target: Exited

- Exited = 1 → churn
- Exited = 0 → non-churn

2️⃣ How To Divide The Data

- Training Set: 80% (for training the model)
- Testing Set: 20% (for evaluating the model)

3️⃣ Models Selected:

- Logistic Regression
- Random Forest Classifier
- Gradient Boosting Classifier
- MLP
- XGBoost/LightGBM/CatBoost

MODEL #1

Logistic Regression

Baseline Model (Threshold = 0.50)

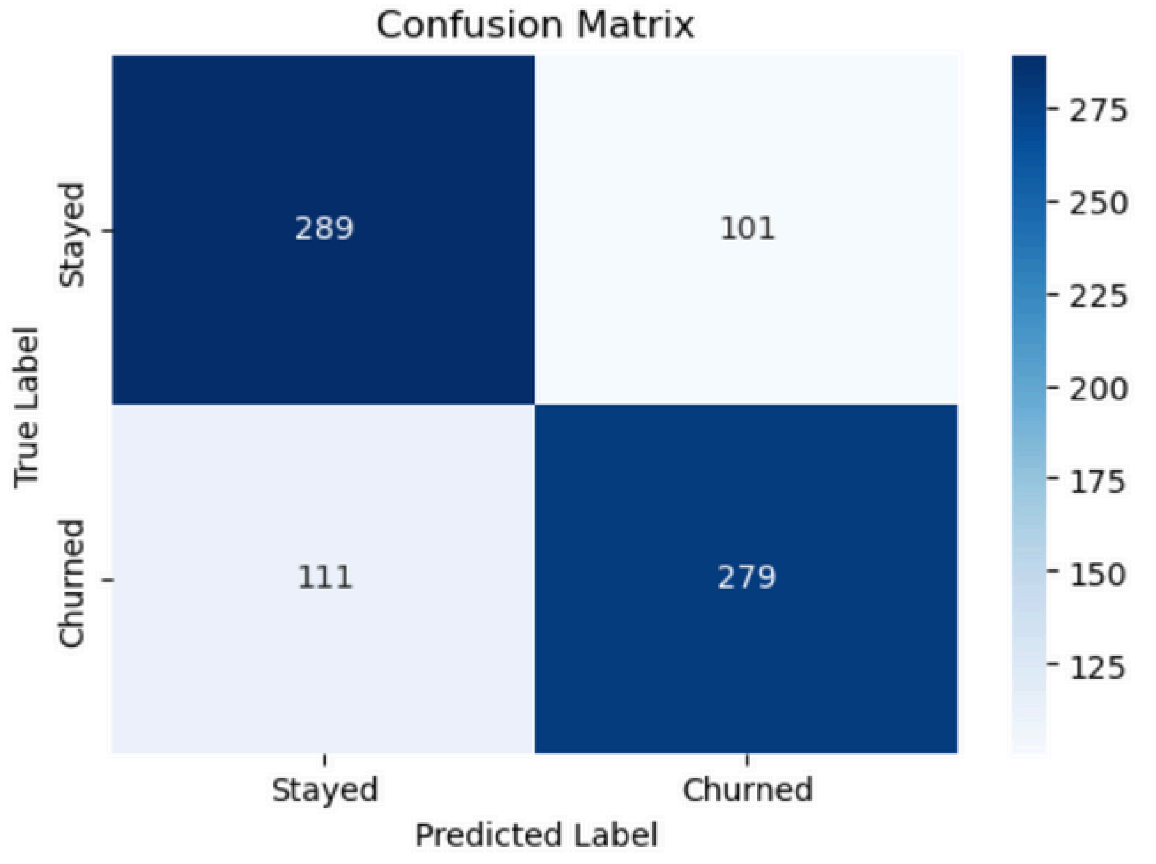
Accuracy: 72.82%

```
Accuracy: 0.7282
ROC AUC Score: 0.7862

Classification Report:
              precision    recall  f1-score   support

           0       0.72      0.74      0.73       390
           1       0.73      0.72      0.72       390

    accuracy                           0.73       780
   macro avg       0.73      0.73      0.73       780
weighted avg       0.73      0.73      0.73       780
```



ROC AUC score for both models: 0.7862
***This value is independent of the chosen classification threshold**

Optimized Model (Threshold = 0.47)

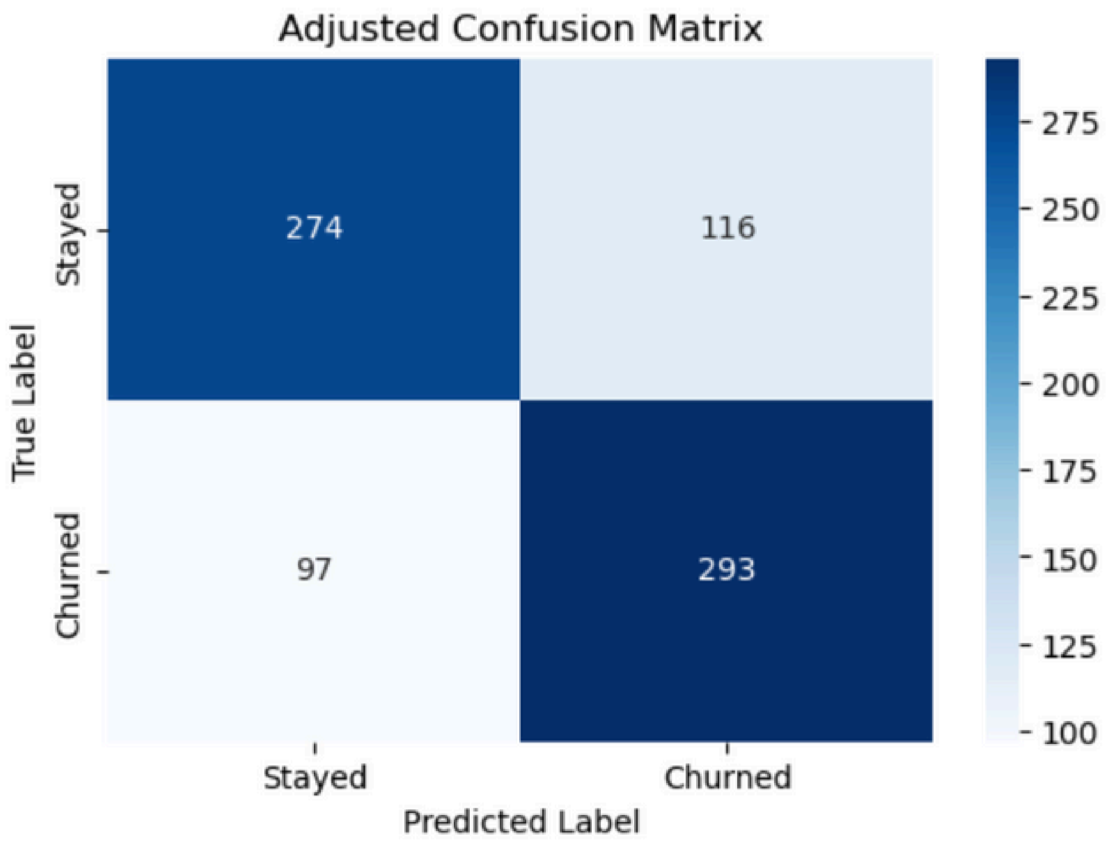
Accuracy: 72.69%

```
Best threshold for decision: 0.47
Adjusted Accuracy: 0.7269
ROC AUC Score: 0.7862

Adjusted Classification Report:
              precision    recall  f1-score   support

           0       0.74      0.70      0.72       390
           1       0.72      0.75      0.73       390

    accuracy                           0.73       780
   macro avg       0.73      0.73      0.73       780
weighted avg       0.73      0.73      0.73       780
```



MODEL #1

Logistic Regression

Baseline Model (Threshold = 0.50)

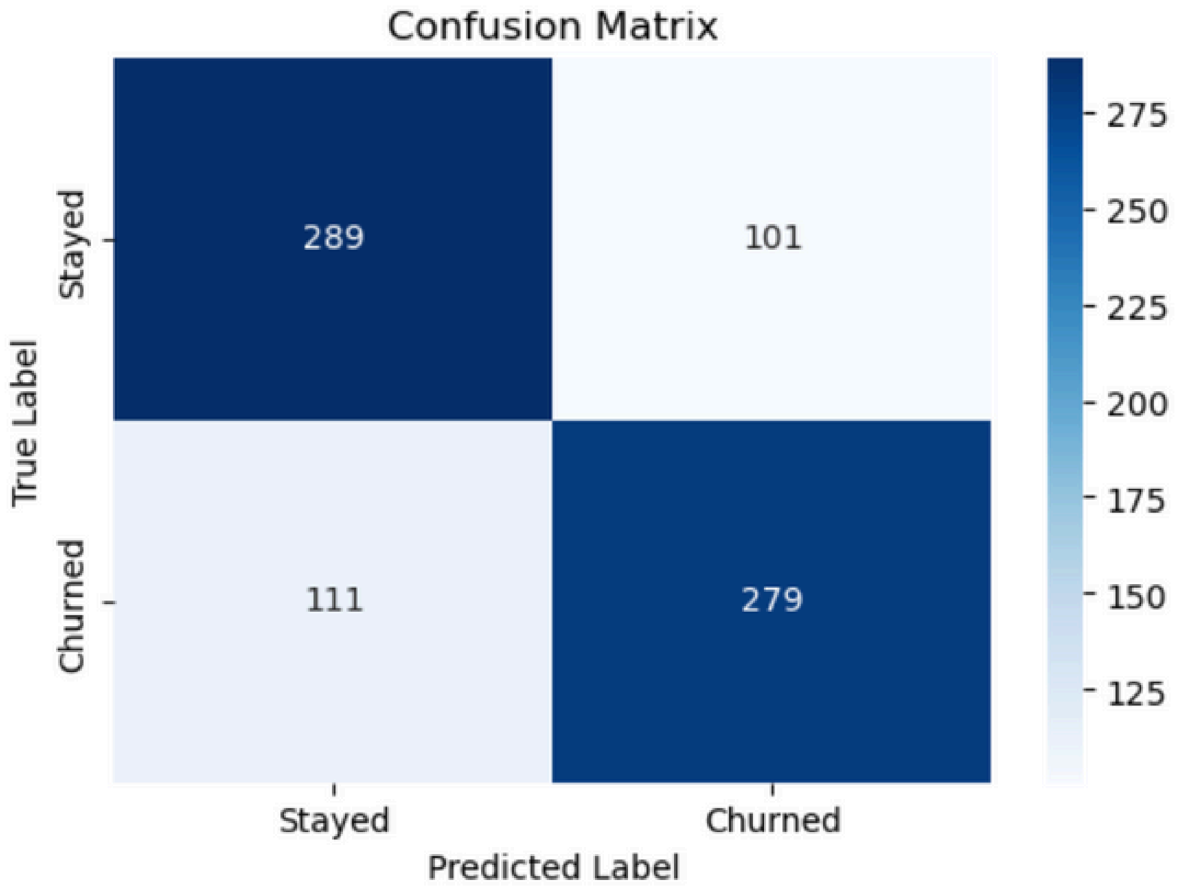
Accuracy: 72.82%

```
Accuracy: 0.7282
ROC AUC Score: 0.7862

Classification Report:
              precision    recall  f1-score   support

           0       0.72      0.74      0.73       390
           1       0.73      0.72      0.72       390

    accuracy                           0.73       780
   macro avg       0.73      0.73      0.73       780
weighted avg       0.73      0.73      0.73       780
```



Insights

Strengths of Baseline Model

- Balanced Recall (72%) & Precision (73%) → The model correctly identifies churners and non-churners at a fairly even rate.
- Good Ranking Ability (ROC AUC = 0.7862) → The model effectively distinguishes between churners and non-churners.
- Lower false positives (101) → Fewer loyal customers are incorrectly predicted as churners, reducing unnecessary retention costs.

Weakness of Baseline Model

- Misses 111 churners → Some customers who are at risk of churning are not identified.

MODEL #1

Logistic Regression

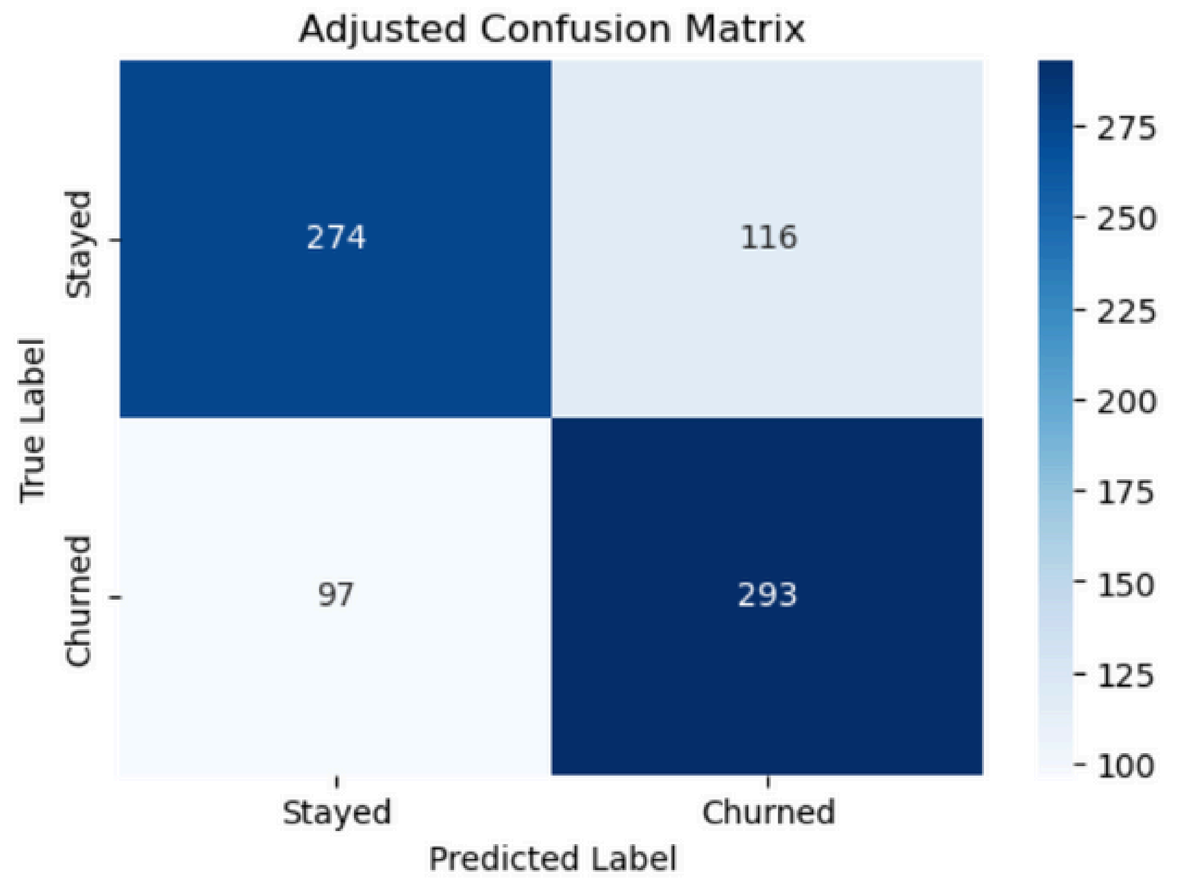
Optimized Model (Threshold = 0.47)
Accuracy: 72.69%

```
Best threshold for decision: 0.47
Adjusted Accuracy: 0.7269
ROC AUC Score: 0.7862

Adjusted Classification Report:
              precision    recall  f1-score   support

           0       0.74      0.70      0.72       390
           1       0.72      0.75      0.73       390

    accuracy                           0.73       780
   macro avg       0.73      0.73      0.73       780
weighted avg       0.73      0.73      0.73       780
```



Insights

🟢 Strengths of Optimized Model

- Higher precision (72% → 74%) → The model is better at correctly identifying actual churners, reducing unnecessary retention efforts.
- False positives reduced (101 → 97) → Fewer loyal customers are mistakenly predicted as churners.
- More balanced recall (75%) → The model captures a higher percentage of actual churners compared to the baseline.

🔴 Weaknesses of Optimized Model

- Slightly lower recall for non-churners (70%)
- Trade-off in accuracy (72.69%) → The model is more precise but does not significantly improve overall accuracy,

MODEL #1

Logistic Regression

Final Insights

- The better choice would the Optimized Logistic Regression Model.
  - Threshold: 0.47
  - Accuracy: 0.7269
  - Recall: 75%
- This model provides the best balance between accuracy and recall.
  - Successfully reduces false positives while maintaining a strong recall.
  - Improved precision in the model helps to focus retention on the right customers.

MODEL #2

Random Forest

```
# Establish pipeline for rfc
pipe_rf = Pipeline([('preprocessing', preprocessor), ('classifier', RandomForestClassifier())])
param_grid = {
    'classifier__max_features': [4, 7, 10, 13, 15],  # Balanced feature selections
    'classifier__max_depth': [4, 7, 10, 13, None]  # Balanced tree depth options
}

grid = GridSearchCV(pipe_rf, param_grid = param_grid, cv = 5, return_train_score=True)
grid.fit(X_train_rfc, y_train_rfc)
```

```
{'classifier__max_depth': 4, 'classifier__max_features': 10}
```

Insights

Notes on Depth and Features

- We had around 18 features in our data (Categorical and Numerical)
- Dropped rows with too many unique values (RowNumber, CustomerId, Surname)
- After dropping, 18-->15 features
- Redundancy in hyperparameters + improving runtime

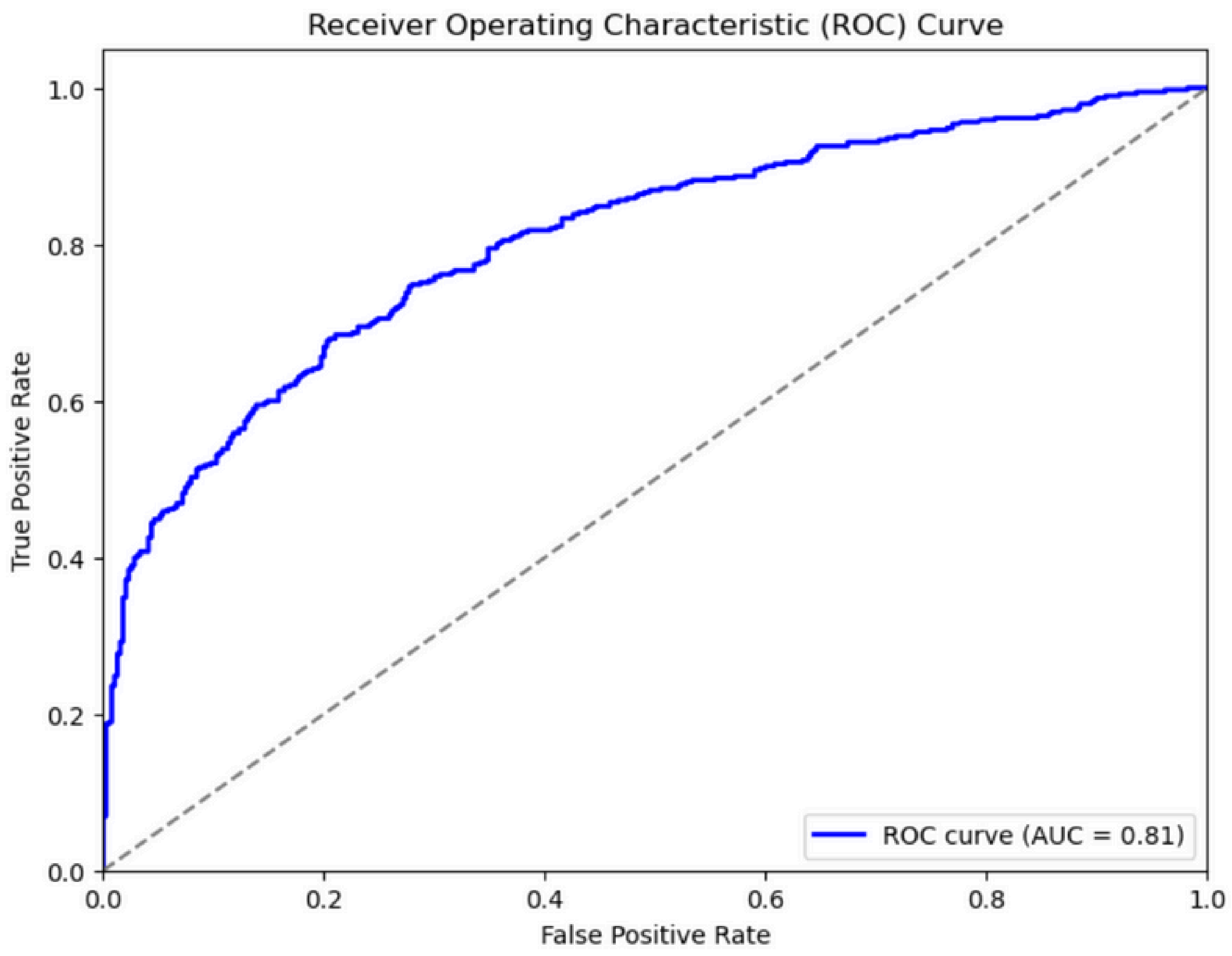
- Model chose the following for optimal depth and features after running the param_grid and GridSearchCV to the left:

Optimal Depth and Features
Depth: 4
Features: 10

MODEL #2

Random Forest

```
Classification Metrics:
Accuracy: 0.7321
Precision: 0.7507
Recall: 0.6949
F1-Score: 0.7217
ROC AUC: 0.8059
```



Insights

🎯 Strengths of Model

- Good Accuracy (73.2%) → The model correctly predicts most customers.
- Good Precision (75.1%) → The model is confident in its churn predictions—most predicted churners actually leave.
- Strong ROC AUC (80.6%) → The model effectively distinguishes churners from non-churners.

⚠ Potential Weaknesses of Model

- Recall (69.5%) → While it's decent, the model still misses some churners, meaning some customers at risk of leaving may not be identified in time.
- F1-Score (72.2%) → While balanced, it suggests there's still room for improvement in recall and precision trade-offs.

MODEL #2

Random Forest

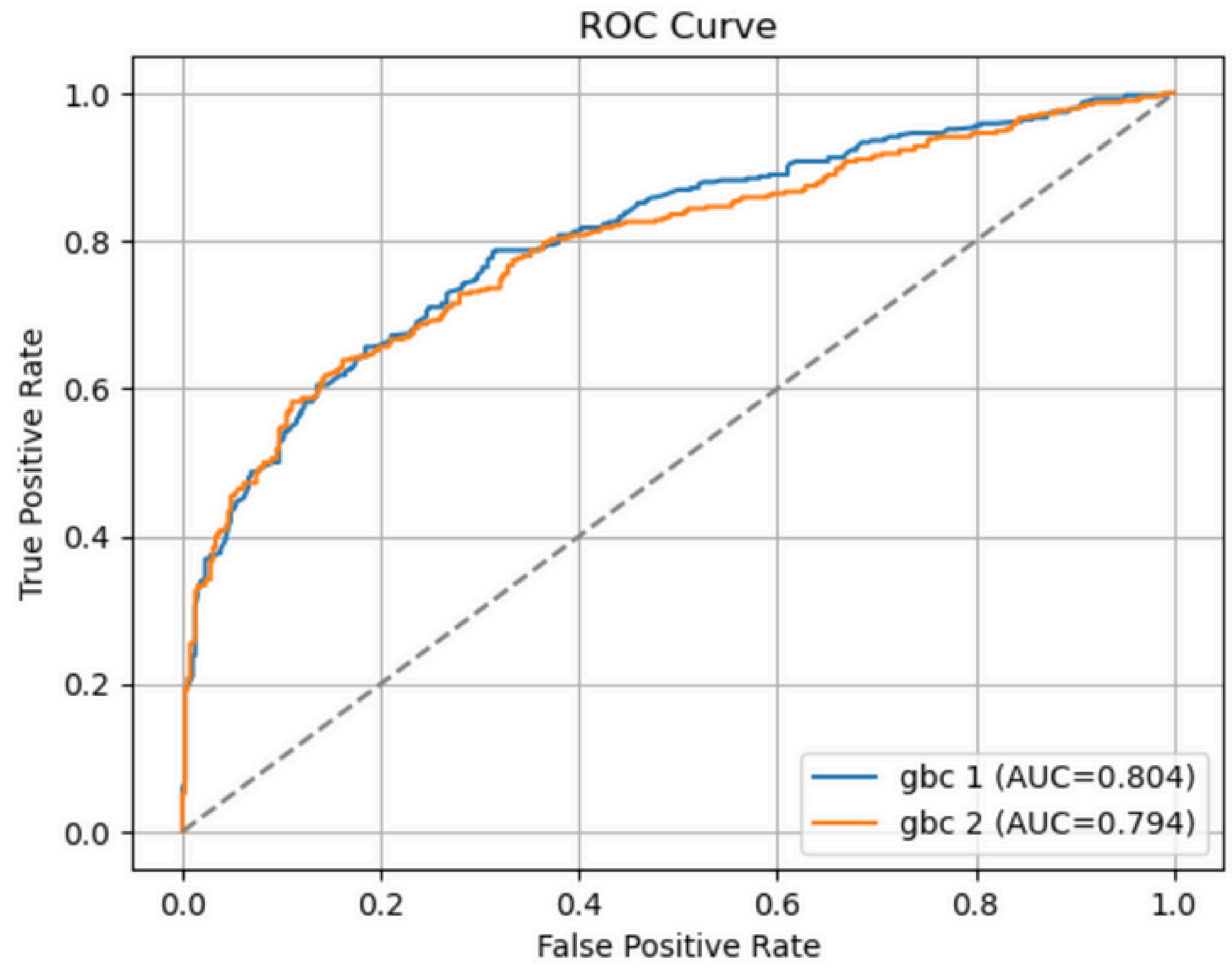
Final Insights

- For this model, its strengths may still have risks:
    - **Precision Risk** (75.1%): It may miss actual churners (low recall) because it avoids making churn predictions unless it's very sure.
- For this model, it's weaknesses have the following risk/notes:
    - **Recall Risk** (69.5%): The bank won't take action in time to retain customers at risk of leaving, leading to lost revenue.
    - **F-1 Notes** (72.2%): It is useful in situations where both false negatives and false positives matter.

✅ Best Practice: High recall is crucial if the goal is to prevent customer churn. If the bank wants to maximize retention efforts, it should focus on models with high recall. Accuracy and other metrics is still important for long-term business goals. This model performs moderately well for accuracy, recall, and precision (73.2%, 69.5%, 75.1%) , as well as balanced F1-Score/ROC AUC (72.2%, 0.8).

MODEL #3

Gradient Boosting Classifier



Model	Accuracy	Precision	Recall	F1 Score
gbc1	0.725641	0.739130	0.697436	0.717678
gbc2	0.717949	0.728495	0.694872	0.711286

1. Decent overall accuracy

2. Decent Precision

- The model is good at identifying the customers who are truly at risk of churning
- The bank can focus retention efforts on this group of customers

3. Slightly Low Recall

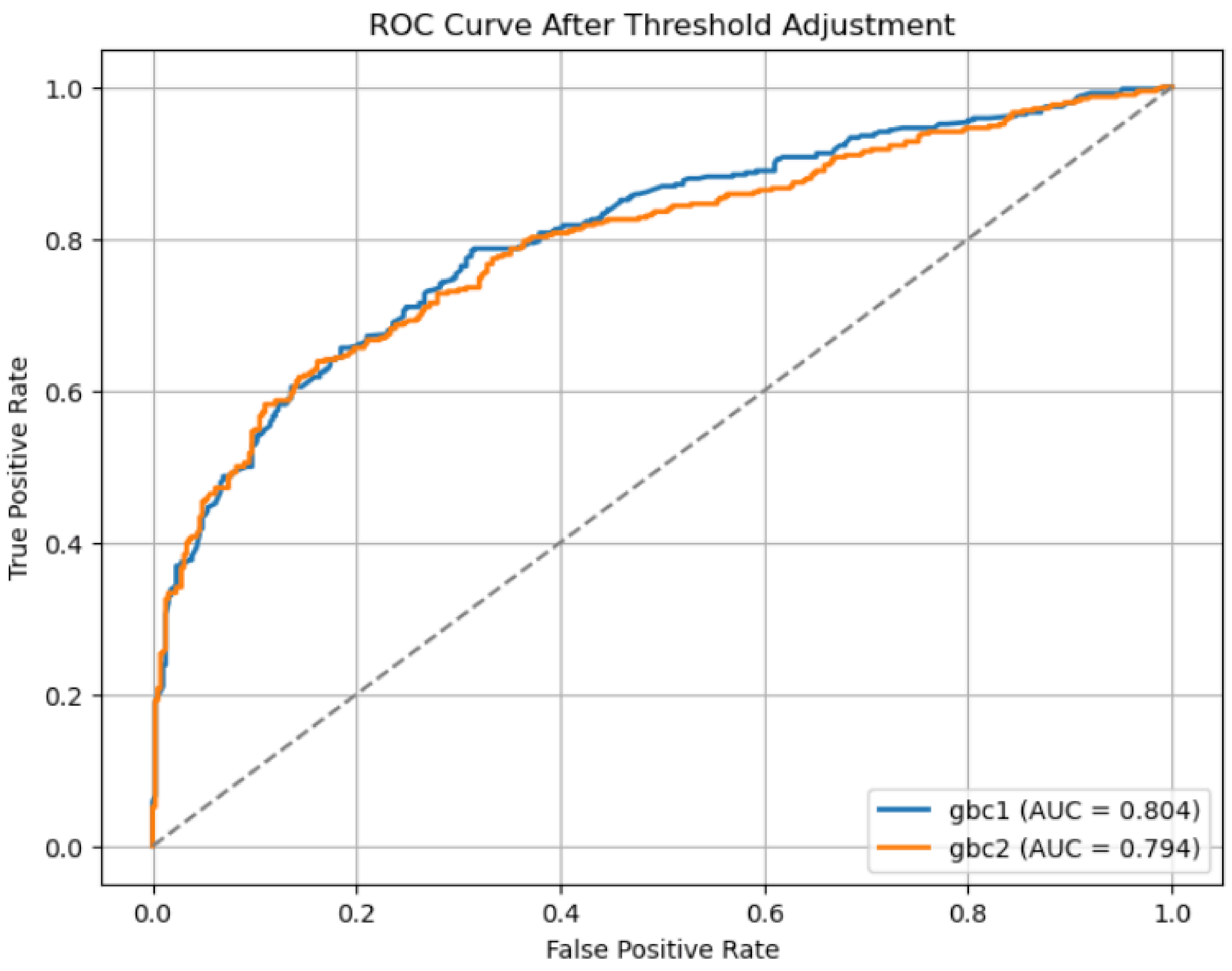
- The model still misses a significant portion of churners
- There is a risk of losing customers without any intervention

4. ROC-AUC Score

- The model has strong ability to distinguish between churners and non-churners

MODEL #3

Gradient Boosting Classifier



1. Slight Decrease in Accuracy

2. Recall vs. Precision Tradeoff

- The model is now better at identifying churners
- Increased FP: can lead to waste the bank's resources

3. F1 Score Improvement

- Better overall balance between precision and recall

4. **Stable ROC-AUC Score**

Model	Best Threshold	Accuracy	Precision	Recall	F1 Score
gbc1	0.424242	0.732051	0.709007	0.787179	0.746051
gbc2	0.393939	0.712821	0.680435	0.802564	0.736471

MODEL #3

Gradient Boosting Classifier

GradientBoostingClassifier

```
class sklearn.ensemble.GradientBoostingClassifier(*, loss='log_loss',
learning_rate=0.1, n_estimators=100, subsample=1.0,
criterion='friedman_mse', min_samples_split=2, min_samples_leaf=1,
min_weight_fraction_leaf=0.0, max_depth=3, min_impurity_decrease=0.0,
init=None, random_state=None, max_features=None, verbose=0,
max_leaf_nodes=None, warm_start=False, validation_fraction=0.1,
n_iter_no_change=None, tol=0.0001, ccp_alpha=0.0)
```

[source]

Source:
https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html

MODEL #4

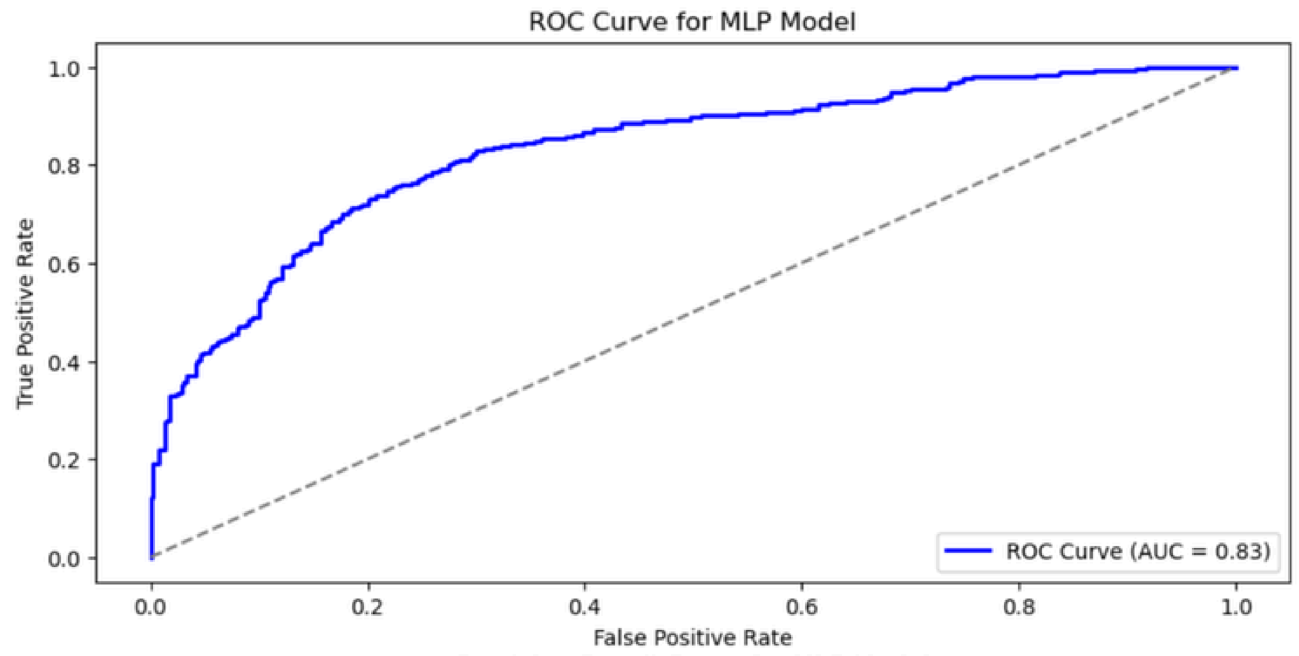
MLP Model

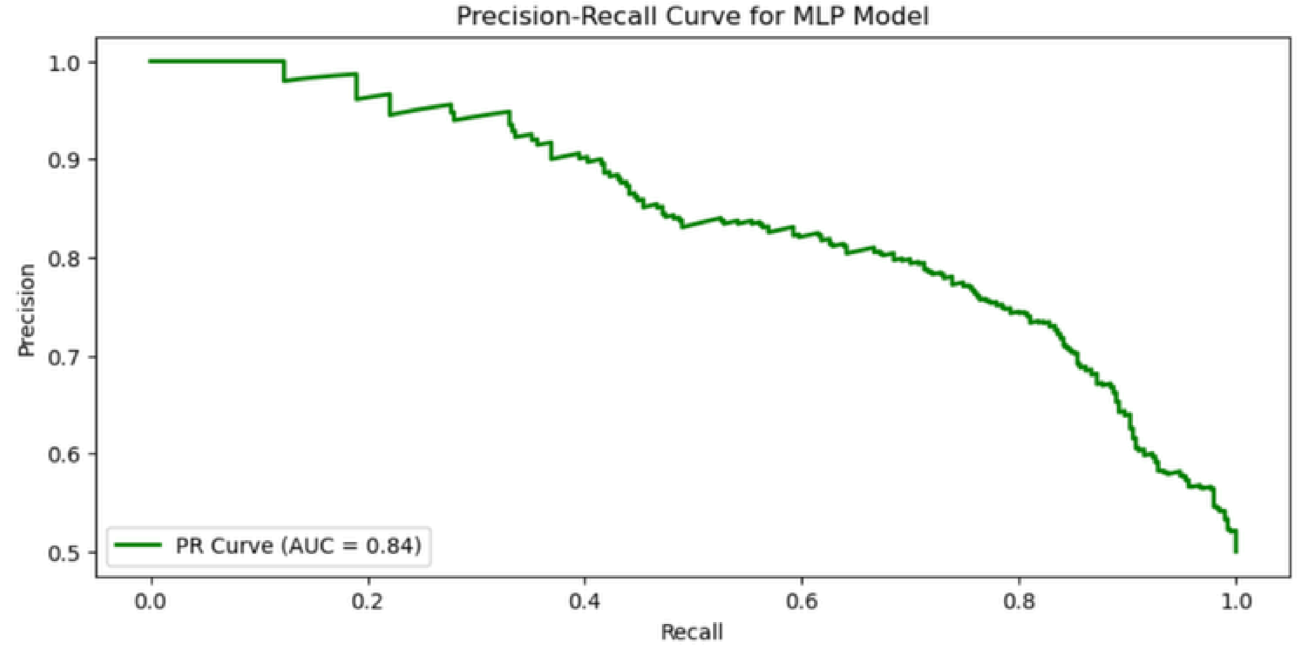
```
MLP Model Accuracy: 0.76
MLP Model Recall for Churners: 0.73
MLP ROC-AUC Score: 0.83

Classification Report:
              precision    recall  f1-score   support

           0       0.75      0.79      0.77       390
           1       0.78      0.73      0.75       390

    accuracy                           0.76       780
   macro avg       0.76      0.76      0.76       780
weighted avg       0.76      0.76      0.76       780
```





⚡ Insights & Recommendations

- The MLP model achieves **76% accuracy** with a **73% recall for churners**, indicating a reasonable ability to detect customers likely to leave.
- The **ROC-AUC score of 0.83** and **PR-AUC score of 0.84** suggest a strong performance in distinguishing between churners and non-churners.
- Despite improvements, the model **still slightly favors non-churners**, suggesting opportunities to further enhance recall without sacrificing precision.
- Next Steps:
  - **Fine-tune cost-sensitive learning** to penalize misclassification of churners instead of relying solely on oversampling.
  - **Experiment with additional regularization techniques**, such as **dropout** or **L2 regularization**, to prevent overfitting.
  - **Test alternative architectures**, including ensemble approaches (e.g., combining MLP with XGBoost) to leverage the strengths of multiple models.

🎯 Conclusion

The MLP model performs well in predicting customer churn, with balanced accuracy and recall. However, **further refinements** through **hyperparameter tuning and class imbalance strategies** could help **maximize recall**, ensuring more effective customer retention strategies.

MODEL #5

WHAT WE GET?

XGBoost/LightGBM/CatBoost

	Accuracy	AUC	F1	Precision	Recall
XGBoost	0.723077	0.802318	0.713528	0.739011	0.689744
LightGBM	0.723077	0.801943	0.713528	0.739011	0.689744
CatBoost	0.723077	0.801473	0.717277	0.732620	0.702564

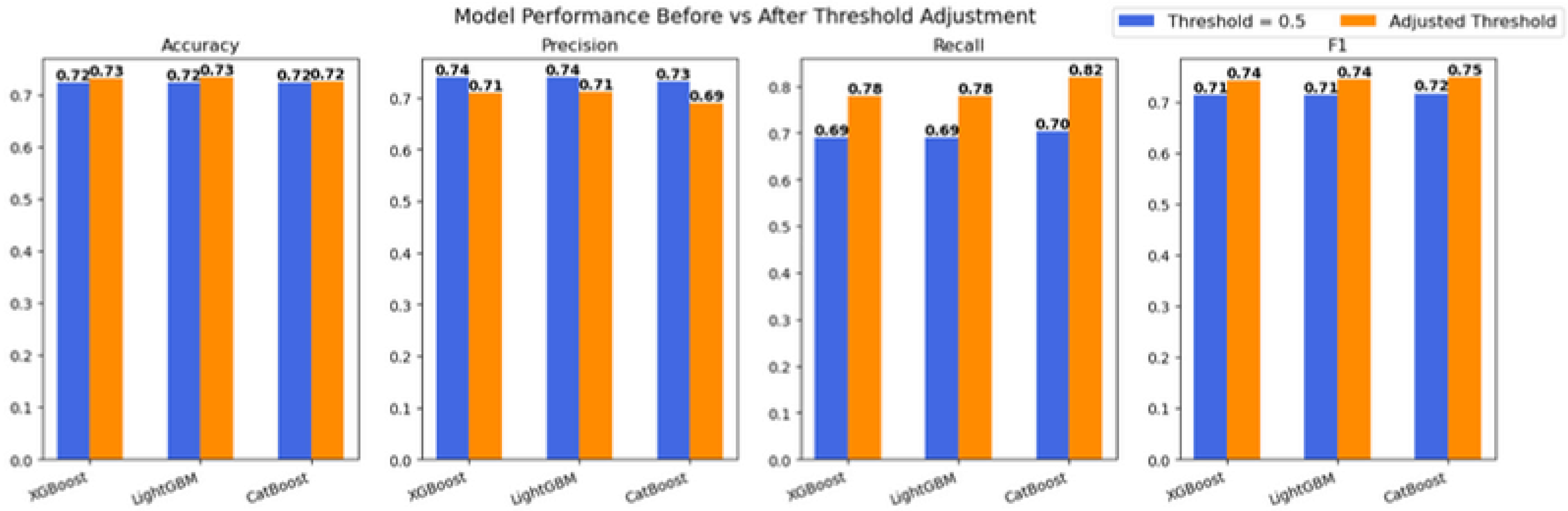
INSIGHTS

• Models show similar accuracy (~0.723) and AUC (~0.80).
• CatBoost has the highest recall (0.703) but still low overall.

⚠ Weaknesses
• Recall is too low (~0.69-0.70) → Many churned customers not detected.
• Business Impact: High false negatives = missed retention opportunities.

🔧 Next Step: Adjust the threshold to improve recall.

INSIGHTS



✅ Key Improvements

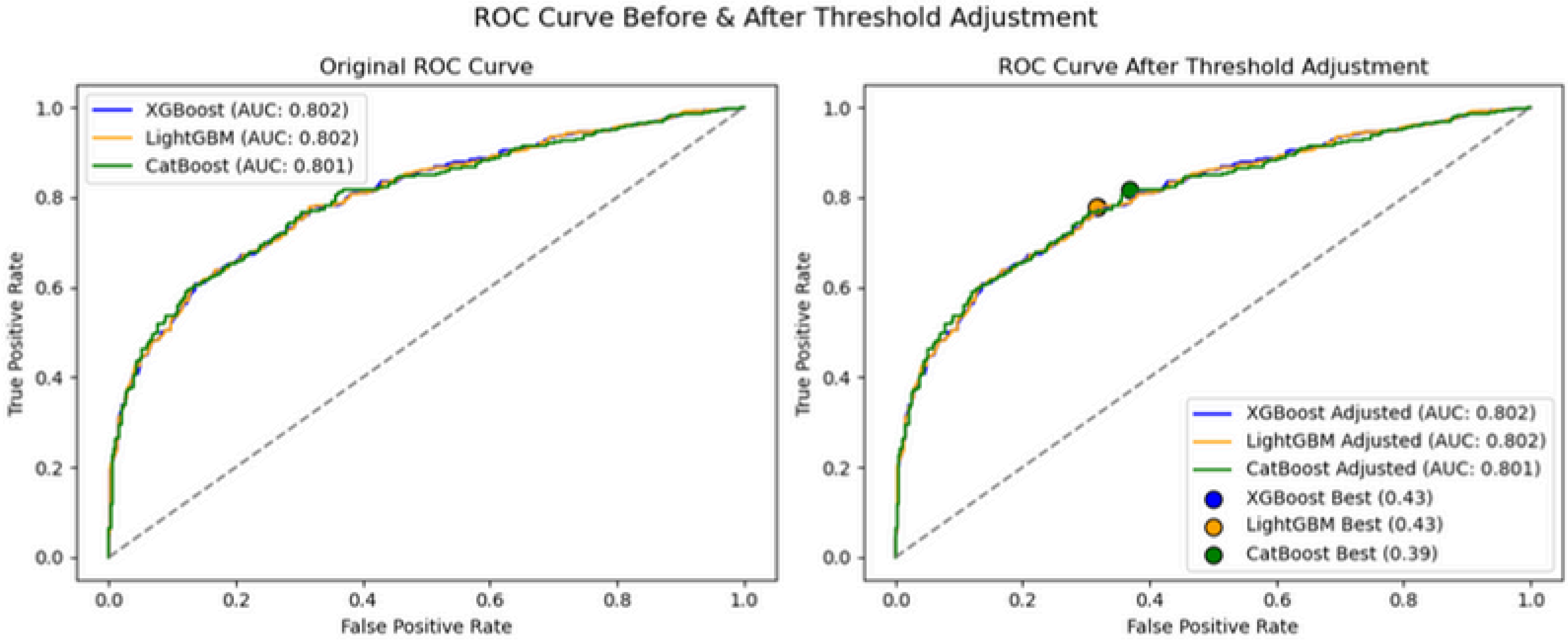
• Recall increased significantly (XGBoost: 0.69 → 0.78, CatBoost: 0.70 → 0.82).

• F1 Score improved, ensuring a better balance between recall & precision.

⚠ Trade-offs

• Precision dropped slightly, meaning more false positives.

• Why it's acceptable? Better to flag potential churners than miss real ones.



✅ Key Findings

- AUC remained stable (~0.80) → Model's performance not compromised.
- Strategic Trade-off: Higher recall without losing discriminatory power.

MODEL #5

XGBoost/LightGBM/CatBoost

Final Insights

- The better choice would the Optimized CatBoost Model
  - Optimized Thresholds: XGBoost (0.43), LightGBM (0.43), CatBoost (0.39)
  - Recall Improvement: XGBoost (0.69 → 0.78), LightGBM (0.69 → 0.78), CatBoost (0.70 → 0.82)
  - Balanced Performance: F1 Score increased while maintaining stable AUC (~0.80)
- Final Takeaways:
  - Threshold tuning significantly improves churn detection, ensuring more at-risk customers are identified.
  - Higher recall enables proactive business actions, helping retain more customers before they leave.
  - Balancing recall & precision is crucial, preventing excessive false positives while improving real-world impact

OPTIMAL MODEL

```
MLP Model Accuracy: 0.76
MLP Model Recall for Churners: 0.73
MLP ROC-AUC Score: 0.83


Classification Report:
              precision    recall  f1-score   support

           0       0.75      0.79      0.77       390
           1       0.78      0.73      0.75       390

    accuracy                           0.76       780
   macro avg       0.76      0.76      0.76       780
weighted avg       0.76      0.76      0.76       780
```

✅ MLP

⚖️ Consistency Across Metrics: The model maintains similar values across accuracy, recall, precision, and F1-score, ensuring no single metric dominates at the cost of another, making it a well-rounded choice for predicting churn.

- 76% Accuracy → Indicates a strong overall classification performance.
- 73% Recall for Churners → Shows a reasonable ability to detect customers likely to leave.
- 76% Overall Recall → Ensures that a good portion of churners and non-churners are correctly identified.
- 76% Precision → Confirms that most predicted churners are actually churners.
- 76% F1-Score → Reflects a well-balanced trade-off between precision and recall.
- 0.83 ROC-AUC Score → Demonstrates strong ability to distinguish between churners and non-churners.

FINAL RECOMMENDATIONS

🔹 **Optimize Decision Thresholds:**

- Adjust dynamically based on business goals:
  - Lower for higher recall (maximize churn detection).
  - Higher for better precision (reduce false positives & intervention costs).

🔹 **Enhance MLP Model Performance:**

- Fine-tune hyperparameters (learning rate, batch size, dropout) for better recall.
- Leverage ensemble methods (MLP + boosting models) to improve churn predictions.
- Expand feature engineering to capture key churn signals (spending, engagement trends).

🔹 **Continuous Monitoring and Adaptation:**

- Retrain regularly with new data.
- Adjust based on market trends & customer behavior shifts.

💡 Business Impact:

- More accurate churn detection → Higher retention & lower costs.
- Smarter decision-making with AI + expert-driven insights.
- Adaptive churn strategy that evolves with the market.

THANK YOU

FOR YOUR ATTENTION

March 10th, 2025